Information Sciences & Technology

Ames Research Center

System Level Verification for Autonomy Software: Analysis of the K9 Rover Executive

NASA Relevance: Verification is essential for autonomy insertion into missions. Software errors are very expensive; traditional testing is hard for autonomous system due to high complexity and unpredictable environments

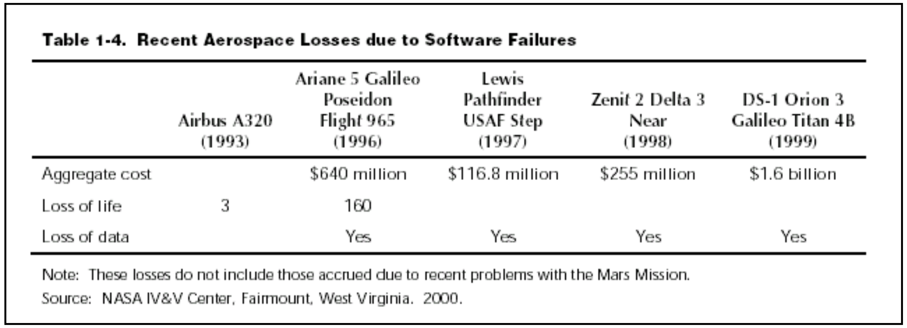
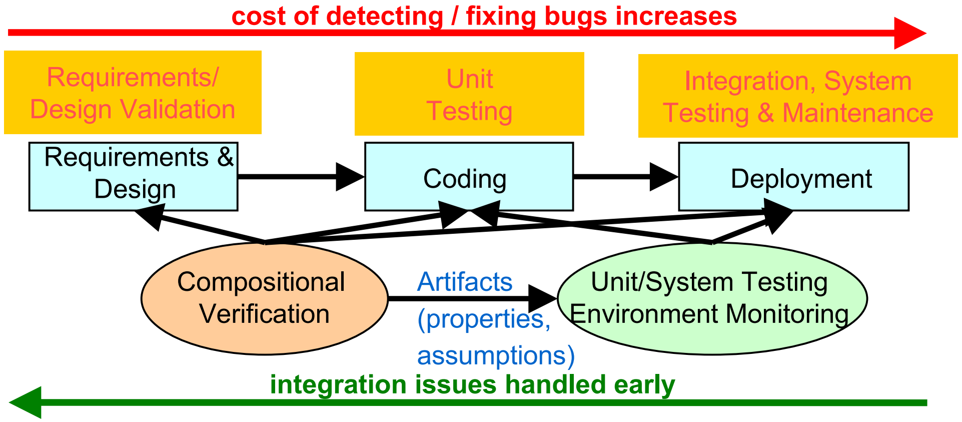


Table 1-4. Recent Aerospace Losses due to Software Failures

	Airbus A320 (1993)	Ariane 5 Galileo Poseidon Flight 965 (1996)	Lewis Pathfinder USAF Step (1997)	Zenit 2 Delta 3 Near (1998)	DS-1 Orion 3 Galileo Titan 4B (1999)
Aggregate cost		$640 million	$116.8 million	$255 million	$1.6 billion
Loss of life	3	160			
Loss of data		Yes	Yes	Yes	Yes

Note: These losses do not include those accrued due to recent problems with the Mars Mission.
Source: NASA IV&V Center, Fairmount, West Virginia. 2000.


cost of detecting / fixing bugs increases
Requirements/ Design Validation
Unit Testing
Integration, System Testing & Maintenance
Requirements & Design
Coding
Deployment
Compositional Verification
Artifacts (properties, assumptions)
Unit/System Testing Environment Monitoring
integration issues handled early

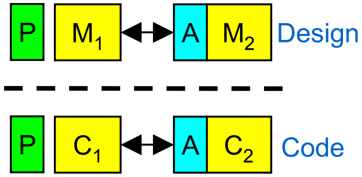
Enabling Technologies:

- To reason about component properties (P), *compositional verification* uses assumptions (A) about component environments
- *Automated frameworks* developed for compositional verification at design level
- *Methodology* developed for re-using design level assumptions for code verification:

- decompose verification problem at code level
- use assumptions to model unimplemented/hardware components
- use assumptions for unit verification and testing (run-time analysis)

Objectives:

- Apply verification tools *in early stages* (design) of software development, when errors are cheaper to detect and fix
- Use design-level artifacts to *guide the implementation* and to enable *more efficient reasoning* at source code level
- Use compositional (*divide and conquer*) techniques that decompose the verification of a system into manageable verification of its components, to achieve scalability with high degree of confidence



Accomplishments:

- Compositional verification applied to autonomy software: **35 KLOC (K9 Rover Executive)**
- Analysis of key properties of the executive (e.g. alternate plan execution) at the design and code level
- Analysis results influenced the developers to **improve** design
- **Design level analysis: 10-1 (memory) improvement** over (non-compositional) model checking
- **Code level analysis: 3-1 (memory) improvement** over (non-compositional) model checking

- Run-time analysis - **improvement** over traditional testing: early detection of integration problems

System Level Verification for Autonomy Software: Analysis of the K9 Rover Executive

- **POC:** Corina Pasareanu (pcorina@email.arc.nasa.gov), Dimitra Giannakopoulou – ASE Group, Code IC
- **Collaborators:** Howard Cannon, Ray Garcia – ARA Group, Code IC; Colin Blundell – University of Pennsylvania
- **Program Funding this Work:** IS
- **Milestone – August'04:** Application of formal methods for the automated verification and validation of a large software system. Successful verification and validation of properties of an autonomy software system on the order of 25KLOC. Focus on concurrency properties. Methodology for detection of errors prior to system integration.
- **Accomplishment / Relation to Milestone:** Development and application of automated verification and validation to the K9 executive - **35 KLOC**, at different stages of software development:
  - Creation and exhaustive analysis of *design models* of the executive, modeling advanced features that allow for *increased autonomy* (e.g. alternate plan execution). Creation of a comprehensive *set of requirements* (both English and formal descriptions) for key concurrency and plan execution properties. Design models and requirements can be successfully *re-used* for design and analysis of future advanced executives. Analysis of design models uncovered several integration problems.
  - Development and application of *methodology* for using design level artifacts for source code verification. Improved performance of code level verification tools (model checking and run-time analysis) by using compositional techniques.
  - Direct impact on new executive design: based on analysis results, the developer created *a new executive*, with simplified architecture to increase modularity and to facilitate reuse.
- **Benefits / Impact of Project:** Development of scalable verification techniques applied early in the software life cycle, when it is cheaper to detect and fix bugs. Compositional techniques that automatically decompose global (system-level) requirements into local properties, which are cheaper – in terms of time and consumed memory – to check. Increased level of confidence in reliability. Early detection of costly integration problems.
- **Future Plans:** Leverage expertise with the analysis of the K9 executive, to participate in the development and analysis of the Plan-Execution Interchange Language (PLEXIL) – funded by MTP. PLEXIL will build on both the K9 Executive concept and the Task Description Language (TDL) Executive, which runs within the CLARAty framework. The PLEXIL Executive will become a generalized Executive within the CLARAty decision layer distribution. In 2005, we plan to continue working on the K9 executive and to start working on the design and analysis of PLEXIL.